IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicants:	Karen L. Noel, et al.	Examiner: Richard Pantolliano
Serial No.:	10/619,853	Group Art Unit: 2194
Filed:	July 15, 2003	Docket No.: 200308870-1
Title:	Method and System of Determining Attributes of a Functional Unit in a Multiple Processor Computer System	

APPEAL BRIEF UNDER 37 C.F.R. § 41.37

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

This Appeal Brief is filed in response to the Final Office Action mailed January 11, 2007, the Notice of Appeal filed on March 12, 2007, and the Notice of Non-Compliant Appeal Brief mailed September 7, 2007.

AUTHORIZATION TO DEBIT ACCOUNT

It is believed that no extensions of time or fees are required, beyond those that may otherwise be provided for in documents accompanying this paper. However, in the event that additional extensions of time are necessary to allow consideration of this paper, such extensions are hereby petitioned under 37 C.F.R. § 1.136(a), and any fees required (including fees for net addition of claims) are hereby authorized to be charged to Hewlett-Packard Development Company's deposit account no. 08-2025.

I. REAL PARTY IN INTEREST

The real party in interest is Hewlett-Packard Development Company, LP, a limited partnership established under the laws of the State of Texas and having a principal place of business at 20555 S.H. 249 Houston, TX 77070, U.S.A. (hereinafter "HPDC"). HPDC is a Texas limited partnership and is a wholly-owned affiliate of Hewlett-Packard Company, a Delaware Corporation, headquartered in Palo Alto, CA. The general or managing partner of HPDC is HPQ Holdings, LLC.

II. RELATED APPEALS AND INTERFERENCES

There are no known related appeals or interferences known to appellant, the appellant's legal representative, or assignee that will directly affect or be directly affected by or have a bearing on the Appeal Board's decision in the pending appeal.

III. STATUS OF CLAIMS

Claims 1 – 20 stand finally rejected. The rejection of claims 1 – 20 is appealed.

IV. STATUS OF AMENDMENTS

No amendments were made after receipt of the Final Office Action. All amendments have been entered.

V. SUMMARY OF CLAIMED SUBJECT MATTER

The following provides a concise explanation of the subject matter defined in each of the claims involved in the appeal, referring to the specification by page and line number and to the drawings by reference characters, as required by 37 C.F.R. § 41.37(c)(1)(v). Each element of the claims is identified by a corresponding reference to the specification and drawings where applicable. Note that the citation to passages in the specification and drawings for each claim element does not imply that the limitations from the specification and drawings should be read into the corresponding claim element or that these are the sole sources in the specification supporting the claim features.

The Summary section of the original specification provides the following summary: In one exemplary embodiment, a method comprises: determining by a first

program an attribute of a first functional unit by referencing a virtual memory address (the first functional unit comprising a first processor and a random access memory (RAM) coupled to the first processor in a computer system, and the first program executing in the first functional unit), determining by a second program an attribute of a second functional unit by referencing the virtual memory address (the second functional unit comprising a second processor and a RAM coupled to the second processor in the computer system, and the second program executing in the second functional unit), wherein referencing the virtual memory address by the first program provides a pointer to an attribute stored in the RAM of the first functional unit, and wherein referencing the virtual memory address by the second program provides a pointer to an attribute stored in the RAM of the second functional unit (see paragraph [0003]).

1. A method comprising:

determining, by a first program, an attribute of a first functional unit (Fig. 1, #s 12, 18, 24) by referencing a virtual memory address, the first functional unit comprising a first processor and a random access memory (RAM) coupled to the first processor in a computer system, and the first program executing in the first functional unit (A processor and RAM form a functional unit: see paragraph [0014]. Each RAD has a page table that provides virtual memory addresses, VMAs, to physical memory addresses, PMAs: see paragraph [0016]. As discussed in paragraphs [0019] – [0021], the RAM in each RAD include pages having programs that are executed by the processes to store and retrieve data.);

determining, by a second program, an attribute of a second functional unit (Fig. 1, #s 12, 18, 24) by referencing the virtual memory address, the second functional unit comprising a second processor and a RAM coupled to the second processor in the computer system, and the second program executing in the second functional unit (A processor and RAM form a functional unit: see paragraph [0014]. Each RAD has a page table that provides virtual memory addresses, VMAs, to physical memory addresses, PMAs: see paragraph [0016]. As discussed in paragraphs [0019] – [0021],

the RAM in each RAD include pages having programs that are executed by the processes to store and retrieve data.); and

wherein the referencing the virtual memory address by the first program provides a pointer to an attribute stored in the RAM of the first functional unit, and wherein the referencing the virtual memory address by the second program provides a pointer to an attribute stored in the RAM of the second functional unit and the pointer and the attribute of the second functional unit are different than the pointer and attribute of the first functional unit (As stated in paragraph [0019], "each VMA A 46, 64 and 80 may thus comprise a pointer to the same physical address in any of read/write pages 56, 72 and 88." As stated in paragraph [0020] "each VMA B 48, 66 and 82 may thus comprise a pointer to physical address for common code pages 58, 74 and 90 respectively." As stated in paragraph [0021], "each VMA C 50, 68, 84 (Figure 2) may thus comprise a pointer to physical address for read-only pages of the RAM of RADs 12, 18 and 24 respectively.").

6. A computer system (Fig. 1, #10) comprising:

a first processor (Fig. 1, #s 16, 22, 26) coupled to a first random access memory (RAM) (Fig. 1, #s 14, 20, 28), the first processor and first RAM forming a first resource affinity domain (RAD) (Fig. 1, #s 12, 18, 24; paragraph [0012]. As stated in paragraph [0015], system resources such as processors and RAM are segmented into functional units called call resource affinity domain (RADs));

a second processor (Fig. 1, #s 16, 22, 26) coupled to a second RAM (Fig. 1, #s 14, 20, 28), the second processor and second RAM forming a second RAD, and wherein the second processor is coupled to the first processor (Fig. 1, #s 12, 18, 24; paragraph [0012]. Figure 1 and paragraph [0012] discuss the processors coupled to RAM and each other.);

a RAD specific attribute of the first RAD along with a replicated portion of an operating system stored in the first RAM (As discussed in paragraphs [0019] and [0020], memory within RAD includes a common code

designation. Common code pages may contain replicated portions of the operating system.); and

a RAD specific attribute of the second RAD along with a replicated portion of the operating system stored in the second RAM(As discussed in paragraphs [0019] and [0020], memory within RAD includes a common code designation. Common code pages may contain replicated portions of the operating system.).

12. A computer readable media comprising an executable program that, when executed, implements a method comprising:

reading a functional unit identifier from a random access memory (RAM) coupled to a processor in which the program executes (As stated in paragraph [0022], RADs store specific attributes such as RAD identifiers that uniquely identify the RAD. As stated in paragraph [0024], a program or portion of the operating system reads the RAD specific attributes from memory);

determining within which functional unit, identified by the functional unit identifier, the program is executing (As stated in paragraph [0024], a program or portion of the operating system reads the RAD specific attributes from memory. "The RAD specific attribute may be, for example, within which RAD the program operating." This paragraph further states that the allocating program determines within which RAD it is operating.); and

addressing data at a same virtual address by different processors in different functional units, wherein each processor in a different functional unit reads different data specific to its functional unit (Paragraphs [0016] and [0018] provide various examples of different processors addressing data at a same virtual address.).

15. A computer system (Fig. 1, #10) comprising:

a first means (Fig. 1, #s 16, 22, 26) for executing programs coupled to a first means (Fig. 1, #s 14, 20, 28) for storing programs and instructions, the first means for executing and means for storing forming a first functional unit (Fig. 1, #s 12, 18, 24; see paragraph [0012] for explanation of recited means being coupled together. As stated in paragraph [0015], system resources such as processors and RAM are segmented into functional units.);

a second means (Fig. 1, #s 16, 22, 26) for executing programs coupled to a second means (Fig. 1, #s 14, 20, 28) for storing programs and instructions, the second means for executing and means for storing forming a second functional unit (Fig. 1, #s 12, 18, 24; see paragraph [0012] for explanation of recited means being coupled together. As stated in paragraph [0015], system resources such as processors and RAM are segmented into functional units.);

an attribute of the first functional unit along with a replicated portion of an operating system stored in the first means for storing (As discussed in paragraphs [0019] and [0020], memory within RAD includes a common code designation. Common code pages may contain replicated portions of the operating system.); and

an attribute of the second functional unit along with a replicated portion of the operating system stored in the second means for storing (As discussed in paragraphs [0019] and [0020], memory within RAD includes a common code designation. Common code pages may contain replicated portions of the operating system.).

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Claims 1 – 5 are rejected under 35 USC § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claims 1, 3, 4, and 12-14 are rejected under 35 USC § 102(e) as being anticipated by US 2003/0088608 (McDonald).

Claim 5 is rejected under 35 USC § 103(a) as being unpatentable over McDonald in view of Boyce.

Claims 2, 6-11, and 15-20 are rejected under 35 USC § 103(a) as being unpatentable over McDonald in view of USPN 6,092,157 (Suzuki).

VII. ARGUMENT

The rejection of claims 1 – 20 is improper, and Applicants respectfully requests withdraw of this rejection.

The claims do not stand or fall together. Instead, Applicants present separate arguments for various claims. Each of these arguments is separately argued below and presented with separate headings and sub-heading as required by 37 C.F.R. § 41.37(c)(1)(vii).

Claim Rejections: 35 USC § 112

Claims 1 – 5 are rejected under 35 USC § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. These rejections are traversed.

The Examiner argues that the recitations in claim 1 are contradictory since the claim "recites that the virtual address accessed by the first and second functional units are the same, and then goes on to state that the pointers are different" (see Final OA at p. 2).

The recitations in claim 1 are accurate and supported in the specification. Figure 2 illustrates four functional units or RADs 12, 18, 24, and 30 with each RAD having a page table 44, 62, 78, and 94. As explained in the specification:

> A page table may be a table, possibly stored in RAM or cache memory of a processor, that may provide virtual memory address (VMA) to physical memory address (PMA) translation. The VMA may be a virtual address used by user and/or operating system programs to access physical memory. In accordance with embodiments of the invention, **each VMA A may have the same virtual address**; each VMA B may have the same virtual address; and each VMA C may have the same virtual address. Depending on the configuration, **VMAs having the same virtual address, however, may not necessarily point to the same physical address**. (Emphasis added: see paragraph [0016]).

The confusion appears to arise from the Examiner not understanding that the page tables associated with each RAD contain differences that allow a virtual address to reference different physical memory based on which functional execution unit makes the reference. In order for a functional unit to reference physical memory, the functional unit first translates the virtual address to a physical address by using the page tables. In prior systems, a single set of page tables existed thus the same virtual address on any functional unit references the same physical memory. The present application allows there to be parts of the page tables that are unique to each RAD. As such, **virtual memory addresses can point to different physical memory depending on what RAD references the virtual address.**

Figure 2 of the present application provides as example of how addressing works in the RADs. By way of example, each RAD references three different virtual memory addresses (VMA A, VMA B, and VMA C). For VMA A, this references read/write memory: all functional units reference the same physical memory. For VMA B, this virtual address references computer instructions within memory. These are read only references by the functional units. Here we see that the code has been replicated into physical memory of the three RADs that contain physical memory. The page tables within the three RADs with physical memory are slightly different and result in translating VMA B to reference the physical memory within these RADs. RAD 30 does not contain any physical memory and uses the page tables associated with RAD 12 to translate the virtual memory address and thus references the physical memory in RAD 12. Since each functional unit with physical memory references local memory, this results in improved performance. The contents of the RAM are identical in this particular case. VMA C represents a virtual memory address that points to a RAD specific attribute. In this case, the contents of the physical memory within each RAD are different. This allows each RAD with physical memory to make a local memory reference to the same VMA to obtain RAD specific data. Without such a mechanism, referencing RAD specific data would first require the processor to lookup the VMA of where the RADs local data is stored and then reference that VMA. Having a common VMA that reference RAD specific data contained in local memory if present results in

increased performance due to not needing to lookup the VMA and by referencing the local memory.

So, the Examiner makes an incorrect assumption that the virtual memory addresses are the pointers to physical memory. As stated above, there is first a translation from the virtual address to the physical memory address through the page tables. The differences within these page tables within the different RADs are what allow the same virtual address to reference different physical memory.

Applicants respectfully ask the Board of Appeals to reverse the rejection under USC § 112, second paragraph.

Claim Rejections: Art Based Rejections

In the final office action, the Examiner makes the following statement: "For the purpose of examination, the Examiner will consider the claim as though the virtual addresses and pointers are the same" (see Final OA at p. 2). As noted above, **virtual memory addresses can point to different physical memory depending on what RAD references the virtual address.** The Examiner has thus misinterpreted the language of the claims. For at least this reason, the art based rejections are flawed since the Examiner has not properly interpreted the actual language of the claims.

Claim Rejections: 35 USC § 102(e)

Claims 1, 3, 4, and 12-14 are rejected under 35 USC § 102(e) as being anticipated by US 2003/0088608 (McDonald). These rejections are traversed.

A proper rejection of a claim under 35 U.S.C. §102 requires that a single prior art reference disclose each element of the claim. See MPEP § 2131, also, *W.L. Gore & Assoc., Inc. v. Garlock, Inc.*, 721 F.2d 1540, 220 U.S.P.Q. 303, 313 (Fed. Cir. 1983). Since McDonald neither teaches nor suggests each element in the claims, these claims are allowable over McDonald.

McDonald generally describes a common ready queue. In other words, the queue resides in memory accessible to all CPUs. Their queue is addressed at the same virtual location on all CPUs. When these virtual addresses are read, **the same data is presented to each CPU**. Further, McDonald discusses thread specific information, as in FIG. 5.

There are attributes in this structure, but they are attributes of the thread, not the CPU or RAD. Further, the thread structure in McDonald is addressed at the same address by all CPUs because it is a common ready queue. Accessing common memory and structures, as in McDonald, slows down the system because of the latency of the NUMA memory.

In contrast to McDonald, one exemplary embodiment of Applicants' invention describes data that are addressed at the same virtual addresses by different CPUs in different RADs. Each CPU in a different RAD sees different data, specific to its RAD. As such, a CPU can quickly determine the attributes of the RAD it is part of.

Claim 1 recites a first functional unit (i.e., a first RAD) that executes a first program referencing a virtual address. Claim 1 then recites a second functional unit (i.e., second RAD) that executes a second program **referencing the same virtual address**. Claim 1 then recites the following:

> wherein the referencing the virtual memory address by the first program provides a pointer to an attribute stored in the RAM of the first functional unit, and wherein the referencing the virtual memory address by the second program provides a pointer to an attribute stored in the RAM of the second functional unit and the pointer and the attribute of the second functional unit are different than the pointer and attribute of the first functional unit..

McDonald does not teach or even suggest these recitations. In fact, as noted above, McDonald works in a completely different way.

For at least these reasons, independent claim 1 and its dependent claims are allowable over McDonald.

Claim 12 also recites numerous recitations that are not taught or even suggested in McDonald. By way of example, claim 12 recites "addressing data at a same virtual address by different processors in different functional units, wherein each processor in a different functional unit reads different data specific to its functional unit." As noted above, McDonald works in a completely different way.

For at least these reasons, independent claim 12 and its dependent claims are allowable over McDonald.

Claim Rejections: 35 USC § 103(a)

Claim 5 is rejected under 35 USC § 103(a) as being unpatentable over McDonald in view of Boyce. These rejections are traversed.

As noted, McDonald does not teach or even suggest all the elements of independent claim 1. Boyce fails to cure the deficiencies of McDonald. Thus, for at least the reasons given in connection with independent claim 1, dependent claim 5 is allowable over McDonald and Boyce.

Claim Rejections: 35 USC § 103(a)

Claims 2, 6-11, and 15-20 are rejected under 35 USC § 103(a) as being unpatentable over McDonald in view of USPN 6,092,157 (Suzuki). These rejections are traversed.

To establish a prima facie case of obviousness, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art cited must teach or suggest all the claim limitations. *See* M.P.E.P. § 2143. For at least the following reasons, Applicants assert that the rejection does not satisfy these criteria.

Independent claims 6 and 15 recite numerous recitations that are not taught or suggested in McDonald and Suzuki. By way of example, claim 6 recites two different RADs, each RAD having a RAM. The claim then recites that each RAD has a replicated portion of the operating system stored in the RAM. In other words, each RAD includes a "**replicated portion**" of the operating system in memory. The Office Action argues that McDonald teaches these recitations at paragraph [0043]. Applicants respectfully disagree.

Paragraph [0043] of McDonald discusses FIG. 3. This figure shows the operating system level 302 that is above the hardware level 301. Paragraph [0043] provides a general discussion of functions for the operating system. Nowhere does paragraph [0043]

state or even suggest that each RAD has a RAM that includes a "replicated portion" of the operating system. Again, paragraph [0043] in McDonald merely provides an overview of functions that an operating system performs.

For at least these reasons, independent claims 6 and 15 and their dependent claims are allowable over McDonald and Suzuki.

CONCLUSION

In view of the above, Applicants respectfully request the Board of Appeals to reverse the Examiner's rejection of all pending claims.

Any inquiry regarding this Amendment and Response should be directed to Philip S. Lyren at Telephone No. 832-236-5529. In addition, all correspondence should continue to be directed to the following address:

Hewlett-Packard Company
Intellectual Property Administration
P.O. Box 272400
Fort Collins, Colorado 80527-2400

Respectfully submitted,

/Philip S. Lyren #40,709/

Philip S. Lyren
Reg. No. 40,709
Ph: 832-236-5529

VIII. Claims Appendix

1. A method comprising:

determining, by a first program, an attribute of a first functional unit by referencing a virtual memory address, the first functional unit comprising a first processor and a random access memory (RAM) coupled to the first processor in a computer system, and the first program executing in the first functional unit;

determining, by a second program, an attribute of a second functional unit by referencing the virtual memory address, the second functional unit comprising a second processor and a RAM coupled to the second processor in the computer system, and the second program executing in the second functional unit; and

wherein the referencing the virtual memory address by the first program provides a pointer to an attribute stored in the RAM of the first functional unit, and wherein the referencing the virtual memory address by the second program provides a pointer to an attribute stored in the RAM of the second functional unit and the pointer and the attribute of the second functional unit are different than the pointer and attribute of the first functional unit.

2. The method as defined in claim 1 wherein, prior to the determining steps, the method further comprises:

saving a replicated portion of an operating system program and the attribute of the first functional unit to a read-only portion of the RAM of the first functional unit; and

saving a replicated portion of an operating system program and the attribute of the second functional unit to a read-only portion of the RAM of the second functional unit.

3. The method as defined in claim 1 wherein determining an attribute of a first functional unit further comprises determining a functional unit identification number.

4. The method as defined in claim 1 wherein determining an attribute of a first functional unit further comprises determining low and high physical address of the RAM of the first functional unit.

5. The method as defined in claim 1 wherein determining an attribute of a first functional unit further comprises determining a list of input/output devices local to the first functional unit.

6. A computer system comprising:

a first processor coupled to a first random access memory (RAM), the first processor and first RAM forming a first resource affinity domain (RAD);

a second processor coupled to a second RAM, the second processor and second RAM forming a second RAD, and wherein the second processor is coupled to the first processor;

a RAD specific attribute of the first RAD along with a replicated portion of an operating system stored in the first RAM; and

a RAD specific attribute of the second RAD along with a replicated portion of the operating system stored in the second RAM.

7. The computer system as defined in claim 6 further comprising:

wherein the replicated portion of the operating system, when executing in the first RAD, reads the RAD specific attribute of the first RAD by reference to a virtual memory address; and

wherein the replicated portion of the operating system, when executing in the second RAD, reads the RAD specific attribute of the second RAD by reference to the virtual memory address.

8. The computer system as defined in claim 6 further comprising:

wherein the RAD specific attribute of the first RAD is a RAD identifier; and

wherein the operating system program stored in the first RAM, when executed by the first processor, determines the RAD within which the operating system program is executed by reading the RAD identifier from the first RAM.

9. The computer system as defined in claim 6 wherein the RAD specific attribute is a RAD identifier.

10. The computer system as defined in claim 9 wherein each replicated operating system program, when executed by the processors in its RAD, uses the RAD identifier to determine a local RAM for memory allocation.

11. The computer system as defined in claim 9 wherein each replicated operating system program, when executed by the processor in its RAD, uses the RAD identifier for scheduling a program stored in local RAM.

12. A computer readable media comprising an executable program that, when executed, implements a method comprising:

reading a functional unit identifier from a random access memory (RAM) coupled to a processor in which the program executes;

determining within which functional unit, identified by the functional unit identifier, the program is executing; and

addressing data at a same virtual address by different processors in different functional units, wherein each processor in a different functional unit reads different data specific to its functional unit.

13. The computer readable media as defined in claim 12 wherein the executable program further comprises allocating memory from RAM within the functional unit, identified by the functional unit identifier, to a program executing on the processor in the functional unit.

14. The computer readable media as defined in claim 12 wherein the executable program further comprises scheduling a program to execute on the processor in the functional unit.

15. A computer system comprising:

a first means for executing programs coupled to a first means for storing programs and instructions, the first means for executing and means for storing forming a first functional unit;

a second means for executing programs coupled to a second means for storing programs and instructions, the second means for executing and means for storing forming a second functional unit;

an attribute of the first functional unit along with a replicated portion of an operating system stored in the first means for storing; and

an attribute of the second functional unit along with a replicated portion of the operating system stored in the second means for storing.

16. The computer system as defined in claim 15 further comprising:

wherein the replicated portion of the operating system, when executing in the first functional unit, reads the attribute of the first functional unit by reference to a virtual memory address; and

wherein the replicated portion of the operating system, when executing in the second functional unit, reads the attribute of the second functional unit by reference to the virtual memory address.

17. The computer system as defined in claim 15 further comprising:

wherein the attribute of the first functional unit is a functional unit identifier; and

wherein the operating system program stored in the first functional unit, when executed by the first means for executing, determines the functional unit within which it is executed by reading the functional unit identifier from the first means for storing.

18. The computer system as defined in claim 15 wherein the attribute is a functional unit identifier.

19. The computer system as defined in claim 18 wherein each replicated operating system program, when executed by the means for executing in its functional unit, uses the functional unit identifier to determine a local means storing for allocation.

20. The computer system as defined in claim 18 wherein each replicated operating system program, when executed by the means for executing in its functional unit, uses the functional unit identifier for scheduling a program stored in a local means for storing.

IX. EVIDENCE APPENDIX

None.

X. RELATED PROCEEDINGS APPENDIX

None.